Simple geometry optimization using the pygeom scripts.

These scripts takes a stream file from CrystFEL as well as the geometry file that was used to create it. It will then match observed positions of strong peaks to their predicted positions after indexing. From this it calculates the shift for each ASIC required to match the two. If there are enough (>50) prediction/peak pairs on an ASIC, it will also try to look at the dependence of the discrepancy between observed and predicted peak position on the x and y laboratory axes to see if the panel can be rotated to improve the match. This latter optimization is very crude, too.

Usage:

python2.7 pygeom_XXX.py [mystream.stream] [mygeom.geom] [number_of_matches]

mystream.stream is the input stream file

mygeom.geom is the input geometry file. All headers and grouping information must be removed with the exception of clen, offset and adu_per_eV. Only an overall offset will be copied to the output.

number_of_matches is the number of peak/prediction pairs to try to obtain from the stream file. If there are less in the stream the script will just take what it can get.

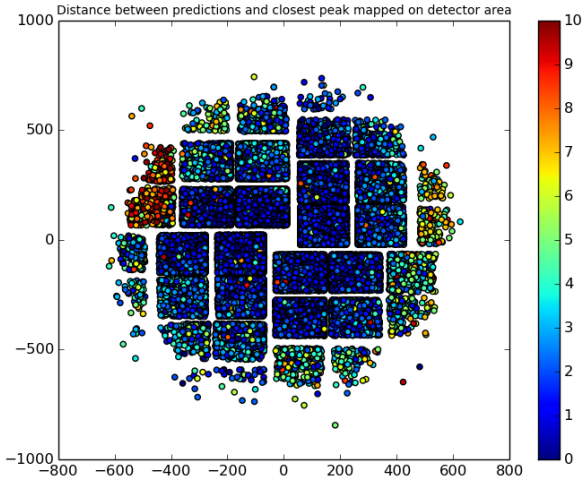
-The script will output a file called output.geom. This will contain the optimized ASIC positions, and some header information that was just copied. You will have to put any other header info into this file yourself.

-The ASIC resolution is hardcoded into the script; pay close attention to this when adapting the script to another detector.

-These scripts are written in python 2.x, not python 3!

-The scripts require numpy and matplotlib

-At the end of the process, the script will display color-coded graphs showing the calculated discrepancies:



Plot of the distanced between observed and predicted peak positions. In the center, the geometry is quite good, but at the edges there are some ASICS whose position is apparently poorly defined. Usually, one or two rounds of refinement will take care of this.